

Aufgabe 1

Zeigen Sie für folgendes Programm P

$x := 5; y := 2; \text{output } (x - (y + \text{read}))$

dass sowohl die operationelle Semantik als auch die Reduktionssemantik bei Eingabe $E = (4)$ die Ausgabe $A = (-1)$ bestimmt.

Wir betrachten die entsprechenden Regeln (aus der Vorlesung) aus beiden Semantiken.

Für die operationelle Semantik:

$$\Delta \langle W|S|I := T.K|E|A \rangle = \langle W|S|T.\text{assign}.I.K|E|A \rangle \quad (\text{OS1a})$$

$$\Delta \langle n.W|S|\text{assign}.I.K|E|A \rangle = \langle W|S[n/I]|K|E|A \rangle, \text{ wobei } n \in \text{ZAHL und}$$

$$S[n/I](x) = \begin{cases} n, & \text{falls } I = x \\ S(x) & \text{sonst} \end{cases} \quad (\text{OS1b})$$

$$\Delta \langle W|S|\text{read}.K|n.E|A \rangle = \langle n.W|S|K|E|A \rangle \text{ für alle } n \in \text{ZAHL} \quad (\text{OS2})$$

$$\Delta \langle W|S|\text{output } T.K|E|A \rangle = \langle W|S|T.\text{output}.K|E|A \rangle \quad (\text{OS3a})$$

$$\Delta \langle n.W|S|\text{output}.K|E|A \rangle = \langle W|S|K|E|n.A \rangle \quad (\text{OS3b})$$

$$\Delta \langle W|S|T_1 \text{ OP } T_2.K|E|A \rangle = \langle W|S|T_1.T_2.\text{OP}.K|E|A \rangle \quad (\text{OS4a})$$

$$\Delta \langle n_2.n_1.W|S|\text{OP}.K|E|A \rangle = \langle (n_1 \text{ OP } n_2).W|S|K|E|A \rangle, \text{ falls } n_1 \text{ OP } n_2 \text{ nicht aus dem darstellbaren Zahlenbereich herausführt} \quad (\text{OS4b})$$

und für die Reduktionssemantik diese Regeln:

$$(x, (s, e, a)) \Rightarrow (s(x), (s, e, a)) \quad , \text{ falls } s(x) \neq \text{frei für } x \in ID, (s, e, a) \in \mathcal{Z} \quad (\text{RS1a})$$

$$(I := T, (s, e, a)) \Rightarrow (\text{skip}, (s[n/I], e', a)) \quad , \text{ falls } (T, (s, e, a)) \xRightarrow{*} (n, (s, e', a)) \quad (\text{RS1b})$$

$$(T_1 \text{ OP } T_2, z) \Rightarrow (n \text{ OP } T_2, z') \quad , \text{ falls } (T_1, z) \xRightarrow{*} (n, z') \quad (\text{RS2a})$$

$$(n \text{ OP } T_1, z) \Rightarrow (n \text{ OP } m, z') \quad , \text{ falls } (T, z) \xRightarrow{*} (m, z') \quad (\text{RS2b})$$

$$(n \text{ OP } n, z) \Rightarrow (n \text{ OP } m, z') \quad , \text{ falls } n \text{ OP } m \in \text{ZAHL} \quad (\text{RS2c})$$

$$\text{read} \Rightarrow (n, (s, e, a)) \quad , \text{ falls } n \in \text{ZAHL} \quad (\text{RS3})$$

$$\text{output } T, (s, e, a) \Rightarrow (\text{skip}, (s, e', a.n)) \quad , \text{ falls } (T, (s, e, a)) \xRightarrow{*} (n, (s, e', a)) \quad (\text{RS4})$$

Durch auflösen der Klammern erhalten wir folgendes Programm im Kontrollkeller

$K = \{x := 5; y := 2; y + \text{read}.x - \text{output}\},$

damit können wir die Kellerspitze simulieren und so die Regeln Schritt für Schritt in folgender Reihenfolge für die operationelle Semantik anwenden:

OS1a, OS1b, OS1a, OS1b, OS4a, OS2, OS4b, OS4a, OS3a, OS3b

Durch induktives anwenden der Regeln für die Reduktionssemantik erhalten das Ergebnis nach anwenden der Regel in dieser Reihenfolge:

RS1b, RS1b, RS3, RS2b, RS2c, RS4

Aufgabe 2

Anmerkung Hinnerk: S hab ich aus der WSKEA Maschine weggelassen, weil es ja eigentlich keinen Speicher und damit kein S gibt. Theoretisch müsste man E und A auch weglassen können, weil es in dieser Aufgabe ja keine Ein-/Ausgabe gibt. Allerdings wäre ich mir dann spätestens bei der Reduktionssemantik nicht mehr sicher... Für c habe ich auch noch nix tolles...

Gegeben sei folgende Syntax:

¹ $W := \text{True} \mid \text{False}$
² $LOP := \text{AND} \mid \text{OR}$
³ $LA := W \mid LA_1 \text{ } LOP \text{ } LA_2 \mid \text{Not } LA$

zur Formalisierung logischer Ausdrücke.

a) Definieren Sie eine geeignete operationelle Semantik.

$z = \langle W | L.K | E | A \rangle$ mit $L \in LA$

$$\Delta \langle W | \underline{true}.K | E | A \rangle := \langle \underline{true}.W | K | E | A \rangle \quad (\text{OS1})$$

$$\Delta \langle W | \underline{false}.K | E | A \rangle := \langle \underline{false}.W | K | E | A \rangle \quad (\text{OS2})$$

$$\Delta \langle W | \underline{not} L.K | E | A \rangle := \langle W | L.\underline{not}.K | E | A \rangle \quad (\text{OS3a})$$

$$\Delta \langle l.W | \underline{not}.K | E | A \rangle := \langle \neg l.W | K | E | A \rangle, \text{ für alle } l \in \{\underline{true}, \underline{false}\}$$

$$\text{wobei } \neg l = \begin{cases} \underline{false} & \text{falls } b = \underline{true} \\ \underline{true} & \text{falls } b = \underline{false} \end{cases} \quad (\text{OS3b})$$

$$\Delta \langle W | LA_1 \text{ } \underline{LOP} \text{ } LA_2.K | E | A \rangle := \langle W | LA_1.LA_2.\underline{LOP}.K | E | A \rangle \quad (\text{OS4})$$

$$\Delta \langle l_2.l_1.W | \underline{OR}.K | E | A \rangle := \langle \underline{true}.W | K | E | A \rangle, \text{ wenn } l_1 = \underline{true} \text{ oder } l_2 = \underline{true} \quad (\text{OS5a})$$

$$\Delta \langle l_2.l_1.W | \underline{OR}.K | E | A \rangle := \langle \underline{false}.W | K | E | A \rangle, \text{ wenn } l_1 = \underline{false} \text{ und } l_2 = \underline{false} \quad (\text{OS5b})$$

$$\Delta \langle l_2.l_1.W | \underline{AND}.K | E | A \rangle := \langle \underline{true}.W | K | E | A \rangle, \text{ wenn } l_1 = \underline{true} \text{ und } l_2 = \underline{true} \quad (\text{OS6a})$$

$$\Delta \langle l_2.l_1.W | \underline{AND}.K | E | A \rangle := \langle \underline{true}.W | K | E | A \rangle, \text{ wenn } l_1 = \underline{false} \text{ und } l_2 = \underline{false} \quad (\text{OS6b})$$

$$\Delta \langle l_2.l_1.W | \underline{AND}.K | E | A \rangle := \langle \underline{false}.W | K | E | A \rangle, \text{ wenn } l_1 = \underline{false} \text{ oder } l_2 = \underline{false} \quad (\text{OS6c})$$

b) Definieren Sie eine geeignete Reduktionssemantik.

$z = (L, (E, A))$ mit $L \in LA$

$$(\underline{not} L, (E, A)) \Rightarrow (\underline{not} L', (E', A)), \text{ falls } (L, (E, A)) \Rightarrow (L', (E', A)) \quad (\text{RS1a})$$

$$(\underline{not} L, (E, A)) \Rightarrow (\neg l, (E, A)), \text{ für } l \in \{\underline{true}, \underline{false}\} \quad (\text{RS1b})$$

$$(L_1 \text{ } \underline{LOP} \text{ } L_2, (E, A)) \Rightarrow (L'_1 \text{ } \underline{LOP} \text{ } L_2, (E', A)), \text{ falls } (L_1, (E, A)) \Rightarrow (L'_1, (E', A)) \quad (\text{RS2a})$$

$$(l \text{ } \underline{LOP} \text{ } L, (E, A)) \Rightarrow (l \text{ } \underline{LOP} \text{ } L', (E', A)), \text{ falls } (L, (E, A)) \Rightarrow (L', (E', A)) \quad (\text{RS2b})$$

$$(l_1 \text{ } \underline{AND} \text{ } l_2, (E, A)) \Rightarrow (eval(l_1 \text{ } \underline{AND} \text{ } l_2), (E', A)), \text{ falls}$$

$$l_1 \text{ } \underline{AND} \text{ } l_2 \text{ ausgewertet werden kann} \quad (\text{RS3})$$

$$(l_1 \text{ } \underline{OR} \text{ } l_2, (E, A)) \Rightarrow (eval(l_1 \text{ } \underline{OR} \text{ } l_2), (E', A)), \text{ falls}$$

$$l_1 \text{ } \underline{OR} \text{ } l_2 \text{ ausgewertet werden kann} \quad (\text{RS4})$$

c) Beweisen Sie die Äquivalenz Ihrer Lösungen zu a) und b).

Lösungsidee von Tob: Wir definieren einen Ausdruck mit der oben genannten Syntax:

¹ $\text{True OR Not (False AND False)}$

(Kontrolle: True)

... und überprüfen mit Struktureller Induktion die beiden Semantiken, ob wir das gewünschte Ergebnis erhalten.

(a) Durch auflösen der Klammerung erhalten wir folgendes Programm im Kontrollkeller K

$$z_0 = \langle W | \text{False AND False.Not.True.OR} | E | A \rangle$$

$$\begin{aligned} z_0 &= \langle \epsilon | \text{False AND False.Not.True.OR} . \epsilon | \epsilon \rangle \\ &\xrightarrow{\text{OS}_4} \langle \epsilon | \text{False.False.AND.Not.True.OR} . \epsilon | \epsilon \rangle \\ &\xrightarrow{\text{OS}_2} \langle \text{False} . \epsilon | \text{False.AND.Not.True.OR} . \epsilon | \epsilon \rangle \\ &\xrightarrow{\text{OS}_2} \langle \text{False.False} . \epsilon | \text{AND.Not.True.OR} . \epsilon | \epsilon \rangle \\ &\xrightarrow{\text{OS}_{6b}} \langle \text{True} . \epsilon | \text{Not.True.OR} . \epsilon | \epsilon \rangle \\ &\xrightarrow{\text{OS}_{3b}} \langle \text{False} . \epsilon | \text{True.OR} . \epsilon | \epsilon \rangle \\ &\xrightarrow{\text{OS}_1} \langle \text{True.False} . \epsilon | \text{OR} . \epsilon | \epsilon \rangle \\ &\xrightarrow{\text{OS}_{5a}} \langle \text{True} . \epsilon | \epsilon | \epsilon \rangle \end{aligned}$$

$$(b) \ z_0 = (\text{True } \underline{\text{Or}} \ \text{Not } \underline{\text{False}} \ \underline{\text{And}} \ \underline{\text{False}}, (E, A))$$

$$\begin{aligned} z_0 &= (\text{True } \underline{\text{Or}} \ \text{Not } \underline{\text{False}} \ \underline{\text{And}} \ \underline{\text{False}}, (\epsilon, \epsilon)) \\ &\xrightarrow{\text{RS}_3} (\text{True } \underline{\text{Or}} \ \text{Not } \text{False}, (\epsilon, \epsilon)) \\ &\xrightarrow{\text{RS}_{1b}} (\text{True } \underline{\text{Or}} \ \text{True}, (\epsilon, \epsilon)) \\ &\xrightarrow{\text{RS}_4} (\text{True}, (\epsilon, \epsilon)) \end{aligned}$$

Aufgabe 3 (freiwillig)

- Implementieren Sie die Reduktionssemantik von WHILE in eine Programmiersprache Ihrer Wahl.
- Implementieren Sie die Semantikfunktion eval, die jeder Programm-Daten-Kombination die entsprechende Ausgabe zuordnet.
- Testen Sie Ihre Funktion eval am Beispiel des ganzzahligen Divisionsprogramms.

Hinweis: Bei Besprechung dieser Aufgabe wird ein Beamer zur Verfügung stehen.